

جامعة طرابلس
كلية تقنية المعلومات
قسم نظم المعلومات

المقرر الدراسي ITGS124

تحليل وتصميم نظم

Systems Analysis and Design

المحاضرة الثانية

«Lecture 2»

أ.فاطمة بشير القاضي
algadyfatma@gmail.com

منهجيات تطوير الأنظمة

تصف دورة حياة تطوير النظم المعلومات المراحل التي تمر بها عملية تطوير النظام (المنظومة). المنهجية هي نهج أو طريقة لتنفيذ مراحل دورة حياة تطوير البرمجيات (انظمة المعلومات).

حيث هناك أنواع كثيرة من مشاريع تطوير نظم المعلومات، وكل مشروع له خصائصه وظروفه التي قد لا تتشابه مع أي مشروع آخر. وفرض هذا الأمر ضرورة وجود أكثر من طريقة (منهجية)، لتناسب كل منها فئة أو مجموعة فئات من المشاريع المتنوعة، وفيما يلي عرض لأهم هذه المنهجيات ولأهم ملامحها.

1. نموذج الشلال Waterfall Model

نموذج (منهجية) الشلال Waterfall Model هي أول الطرق التي جرى استخدامها لتطوير الأنظمة، وهي طريقة من السهل فهمها وتطبيقها. تمر هذه الطريقة بكل مراحل تطوير البرمجيات وبشكل متتابع، بداية من التخطيط وحتى توزيعه لبيئة العمل الفعلية وتنفيذ ما يلزمه من صيانة. حيث أُتت تسمية هذا النموذج بالنموذج الشلال نتيجةً تتالي هذه المراحل مع بعضها البعض بشكلٍ تدفقي باتجاه الأسفل بما يشبه الشلال، فلا تبدأ مرحلة قبل انتهاء المرحلة السابقة لها، وتمثل نتيجة كل مرحلة المُدخل الذي ستعمل عليه المرحلة التي تليها. ولا يمكن أن تتداخل المراحل فيما بينها.

قد يجد مطورو البرمجيات منهجية الشلال هي الأنسب عندما تكون متطلبات النظام واضحة وثابتة من بداية المشروع، فضلاً عن توفر الخبرات اللازمة لإنجاز المشروع من البداية، وعادة ما تستخدم في المشاريع البرمجية الصغيرة.



التخطيط

التحليل

التصميم

التنفيذ

الاختبار والصيانة

تابع نموذج الشلال

ايجابيات نموذج الشلال

- بسيط وسهل الفهم والاستخدام .
- سهل الإدارة نظراً لصلابته، حيث أنه لكل مرحلة نواتج محددة.
- يعمل بشكل جيد في المشاريع الصغيرة والتي تكون متطلباتها مفهومة.
- سهولة ترتيب المهام.
- يتم توثيق العمليات والنتائج بشكل جيد.

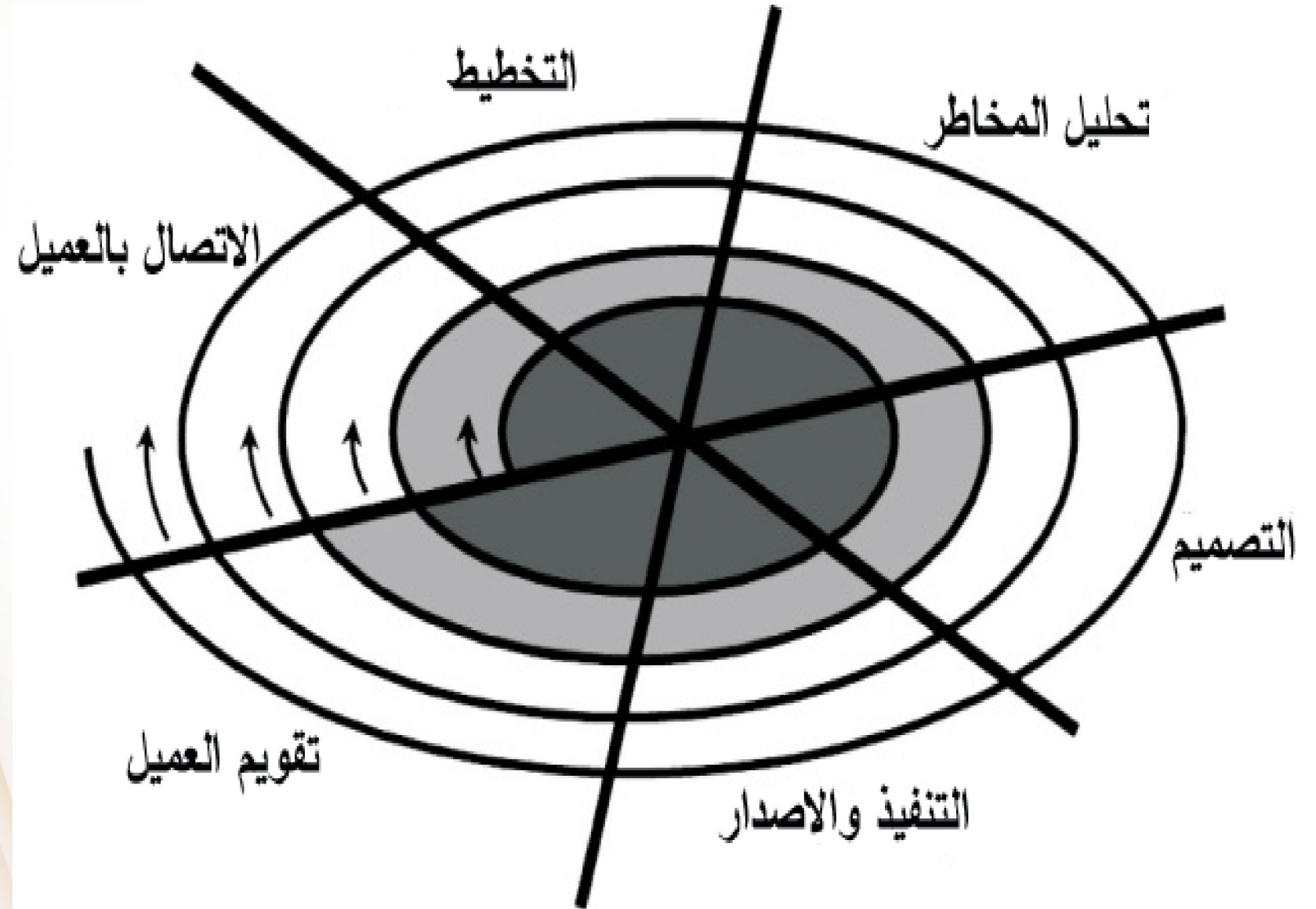
سلبيات نموذج الشلال

- لا يتم إنتاج أي برمجية قابلة للعمل حتى انتهاء دورة الحياة بأكملها.
- كمية كبيرة من المخاطر والشك.
- غير مناسب للمشاريع المعقدة، وللمشاريع ذات المتطلبات المتغيرة.
- فيه ضياع كبير للوقت وذلك لانتظار انتهاء المرحلة السابقة تماماً.
- صعوبة اختبار النظام.

2. نموذج الحلزوني Spiral Model

يعد النموذج الحلزوني أحد أهم نماذج دورة حياة تطوير البرمجيات، والذي يوفر الدعم للتعامل مع المخاطر. إذ يُفضل استخدام هذا النموذج للمشاريع الكبيرة التي تتضمن تحسينات مستمرة.

تبدأ كل مرحلة من مراحل تطوير الانظمة في النموذج الحلزوني بالتصميم وتنتهي بمراجعة العميل للتقدم، إذ تبدأ عملية التطوير في نموذج الحلزوني في دورة حياة تطوير الانظمة بمجموعة صغيرة من المتطلبات وتمر بكل مراحل التطوير لتلك المجموعة من المتطلبات، في كل دورة اللولب المتزايد يقوم فريق تطوير نظم المعلومات بإضافة وظائف للمتطلبات حتى يصبح التطبيق جاهزاً لمرحلة الإنتاج.



تابع نموذج الحلزوني

ايجابيات نموذج الحلزوني

- التطوير المستمر أو المتكرر يساعد في إدارة المخاطر.
- جيد للمشاريع الكبيرة.
- يمكن إضافة وظائف أو إجراء تغييرات إضافية في مرحلة لاحقة.
- هناك دائمًا مساحة لتعليقات العملاء مما يحقق رضا العملاء.

سلبيات نموذج الحلزوني

- صعوبة تقدير الوقت اللازم لإنجاز المشروع.
- تعقيده وكونه مرتفع التكاليف.
- غير مناسب للمشاريع الصغيرة.
- يتطلب خبرة في تقييم المخاطر.

3. التطوير ببناء النماذج الأولية

Prototyping Model

النموذج الأولي هو إحدى منهجيات تطوير البرمجيات يتم فيه بناء النموذج الأولي واختباره وإعادة صياغته حتى يتم الوصول إلى نموذج أولي مقبول. تعمل المنهجية بشكل أفضل في السيناريوهات التي تكون فيها متطلبات المشروع غير معروفة بالتفصيل، وهي طريقة تكرارية وتجريبية حيث تتم بين مطور البرمجيات والعميل.

خطوات بناء النماذج الأولية

1. تحديد المتطلبات.
2. تطوير النموذج الأولي.
3. المراجعة.
4. تحسين وتطوير النموذج.

تحديد المتطلبات المعروفة

بناء النموذج الاول

وضع النموذج تحت
الاستخدام

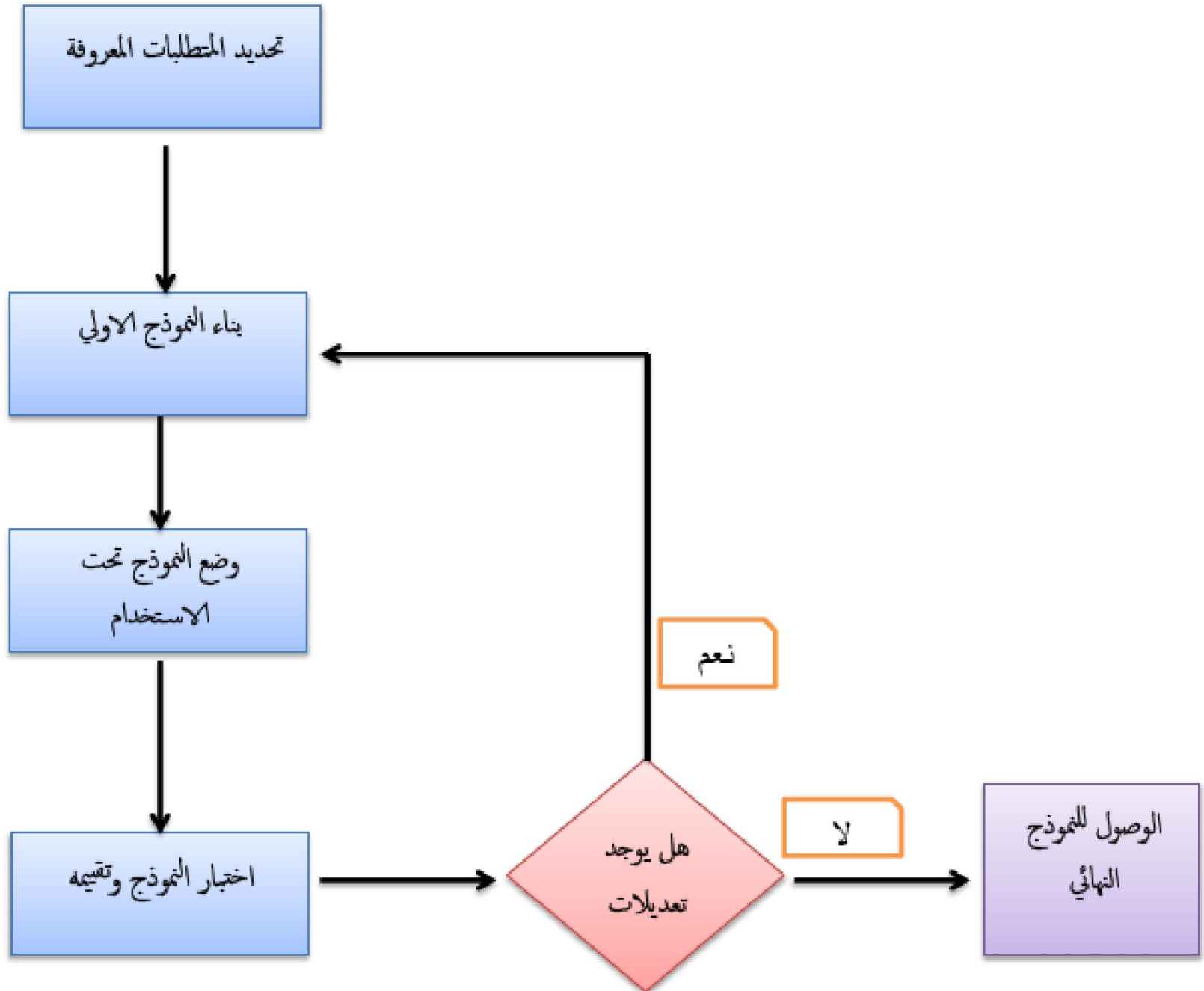
اختبار النموذج وتقييمه

هل يوجد
تعديلات

نعم

لا

الوصول للنموذج
النهائي



تابع التطوير ببناء النماذج الاولى

ايجابيات التطوير ببناء النماذج الاولى

- زيادة جودة التصميم.
- يسهل عملية الصيانة.
- تخفيض الوقت والتكاليف.
- تحسين وزيادة مشاركة المستخدم النهائي.
- ينتج انظمة ترضي المستخدم النهائي.

سلبيات التطوير ببناء النماذج الاولى

- اعطاء وقت محدد لتجربة النظام وقد يكون الوقت غير كافي.
- المستخدم النهائي لا يشارك في مرحلة التطوير.

4. نموذج تطوير البرمجيات الرشيق Agile Model

كلمة Agile عبارة عن صفة وتعريفها من القاموس يعني: التحرك بسرعة وخفة. لا تستند هذه الطريقة على النظريات المجردة بقدر استنادها على تتبع قصص المشروعات الناجحة وخبرات فرق العمل التي قامت عليها. تعمل هذه الطريقة على تجزئة المنتج لعدة عمليات بناء صغيرة تعمل بشكل متكرر وفي كل دورة يكون لدينا منتجاً يعمل ويحمل بعضاً من خصائص المنتج النهائي. تعتمد هذه الطريقة أيضاً على العمل بشكل تراكمي فمع التقدم من دورة إلى التي تليها تتراكم الوظائف والخصائص في المنتج حتى يكتمل المنتج على صورته النهائية كما يريده المستخدمون.

تعتمد طريقة Agile على فرق عمل صغيرة ذات درجة عالية من المهارة وسرعة الاستجابة فضلاً عن استيعاب كافة فرق العمل لهذه الطريقة بشكل ممتاز، كما يتم إشراك عينة من المستخدمين النهائيين داخل الفريق. تعمل Agile بشكل سهل ومرن، وتبدأ التطوير في المنتج المطلوب مباشرة، دون الحاجة لتخطيط مفصل عن مهام العمل أو قائمة كاملة بوظائف وخصائص المنتج النهائي.

تابع نموذج تطوير البرمجيات الرشيقة

ايجابيات نموذج تطوير البرمجيات الرشيقة

- نهج واقعي جدا لتطوير الانظمة.
- يعزز العمل الجماعي والتدريب المتبادل.
- يمكن تطوير الوظائف بسرعة.
- مناسب للمتطلبات الثابتة أو المتغيرة.
- يتطلب تخطيط قليل أو معدوم.
- سهل الادارة.
- وجود حد أدنى من الوثائق التي يتم انشاؤها.

سلبيات نموذج تطوير البرمجيات الرشيقة

- زيادة مخاطر الاستدامة وقابلية الصيانة والتوسعة.
- يعتمد بشكل كبير على تفاعل العميل، لذلك إذا لم يكن العميل واضحا يمكن ان يقود الفرق الى الاتجاه الخاطئ.
- هناك اعتمادية فردية عالية جدا، نظرا لوجود حد أدنى من الوثائق.

